先端人工知能論2 2017年10月10日

深層強化学習

株式会社Preferred Networks
松元 叡一

自己紹介

松元 叡一(@mattya1089)

- 東大理物→総合文化修士→PFN
- 強化学習、ロボット、GAN

株式会社 Preferred Networks

- 2014年3月設立
- 本社：東京
- アメリカに子会社
- 社員数：約100名

- IoT + 人工知能領域にフォーカス
  - 交通
  - 製造業
  - バイオヘルスケア

Chainer

講義の流れ

1. **深層強化学習イントロダクション**
2. 強化学習の問題設定
3. 価値関数ベースの手法 → 11/21の講義で詳しく扱う
4. 方策関数ベースの手法 → 12/19の講義で詳しく扱う

後半にChainer, ChainerRLの紹介と演習を行います

深層強化学習イントロダクション

- 深層学習が強いところ

  →画像、音声、言語などの分類、検出、予測、生成.....

- 行動まで結びつけたい
  - ユーザーのログ(→分類)→表示広告の決定
  - カメラの画像(→人間の検出)→車の回避行動
  - 時刻tまでの値動き(→t+1での値動き予測)→自動取引
  - etc.

深層強化学習イントロダクション

- 行動の学習の難しさと強化学習の問題設定
  - 正解が分からない
    - 「報酬」から正解を間接的に学習する
  - 行動の結果が試行するまで分からない
    - 探索(試行錯誤)してデータを集める
  - 自分の行動が未来に影響を与える
    - 将来における報酬合計を最適化する

OpenAI gymの
Humanoid agent

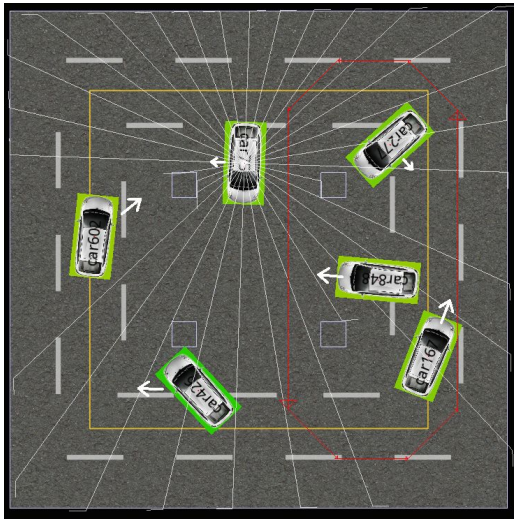
強化学習の位置付け

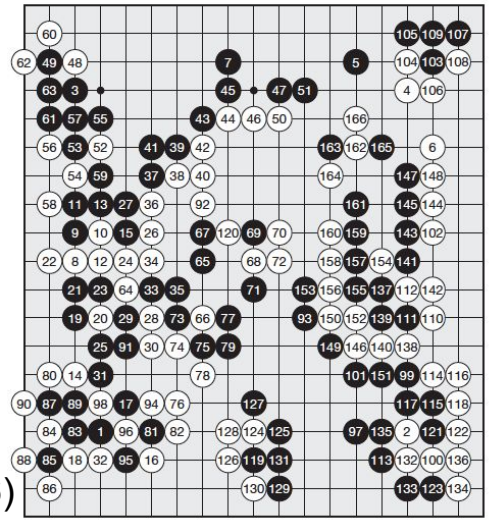
問題に関する
知識・手がかり

多

少

- ルールベース
- ヒューリスティック探索
- 教師あり学習(模倣学習)
- 強化学習
- 進化的手法





(Silver, 2016)

「深層」強化学習の強み

- 深層学習と組み合わせる利点
  - 高次元のデータにもスケールする
  - 少ないドメイン知識で学習できる
  - End-to-endの学習
  - 転移学習

Levine et al. 2016

(a) learn to explore in Level-1 (b) explore faster in Level-2

https://arxiv.org/pdf/1705.05363.pdf

応用例

- ゲームAI(Atari, AlphaGo)
- ロボット
- 自動運転(自動車・ドローン)
- 広告表示
- 電力最適化
- 金融取引

深層強化学習の利用例

講義の流れ

1. 深層強化学習イントロダクション
2. **強化学習の問題設定**
3. 価値関数ベースの手法
4. 方策関数ベースの手法

強化学習の問題設定

環境(Environment)

エージェント(Agent)

状態(State)

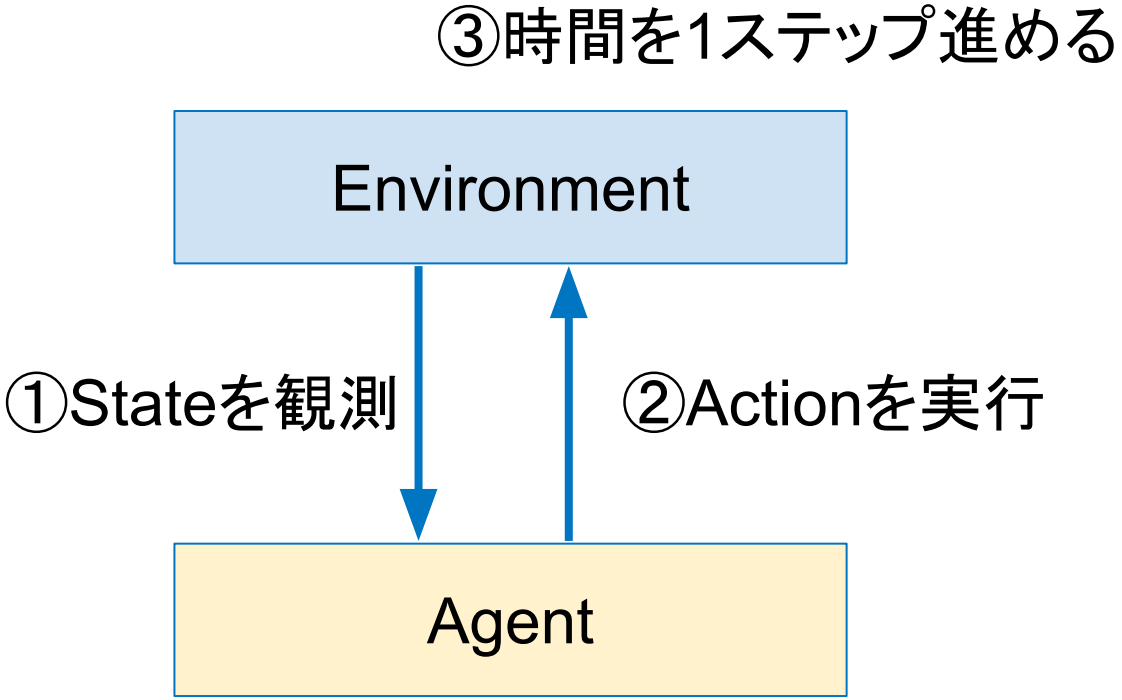
行動(Action)

報酬(Reward)

$$S_t \in \mathcal{S}$$

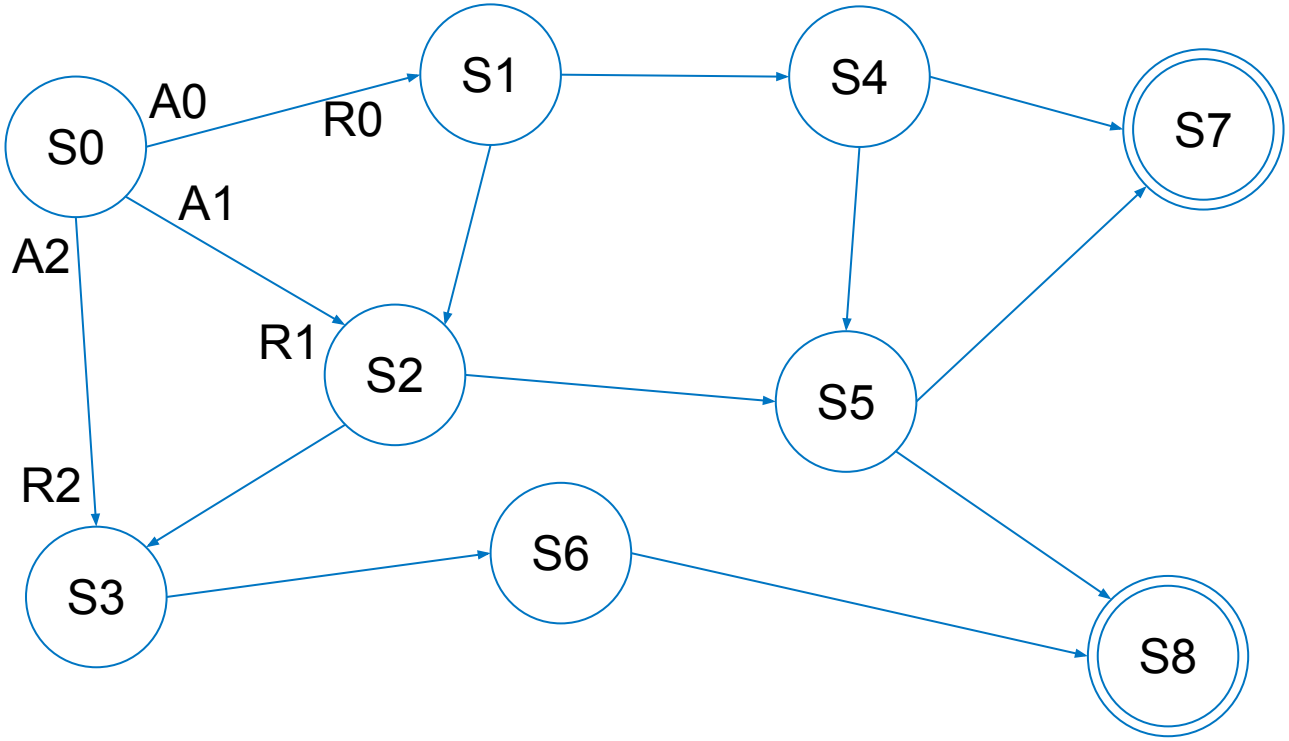
$$A_t \in \mathcal{A}$$

$$R_t \in \mathbb{R}$$



$$S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, \ldots$$

強化学習の問題設定



- 決定的／確率的, 有限状態／無限状態, マルコフ性を満たすか, エージェントは完全な状態を知ることが出来るか, ……といったバリエーションが有る
- 強化学習の理論ではMarkov Decision Processで定式化することが多い

強化学習の問題設定

- 方策(Policy):エージェントの行動を決める関数

$$a = \pi(s)$$ （決定的）

$$P(A_t = a|S_t = s) = \pi(a|s)$$ （確率的）

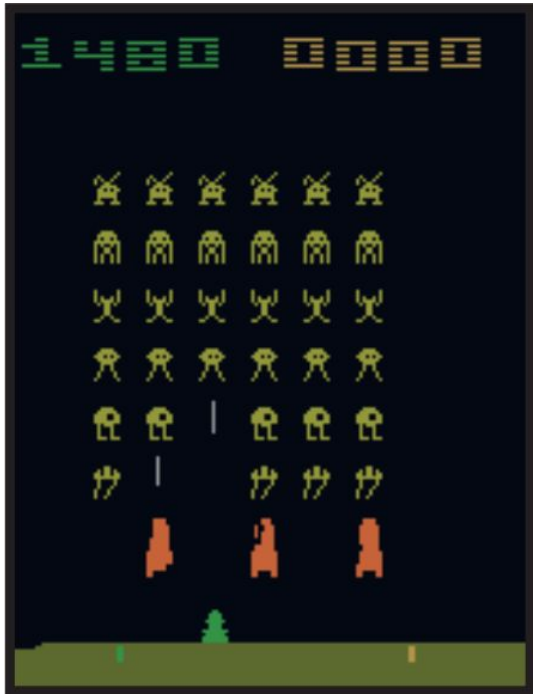
- 利得(Return) ←これを最大化するような方策が知りたい

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$

γ: discount factor $\in$ [0, 1]

強化学習の具体例

- 目標：Atariのゲームで、ゲーム終了時のスコアの最大化



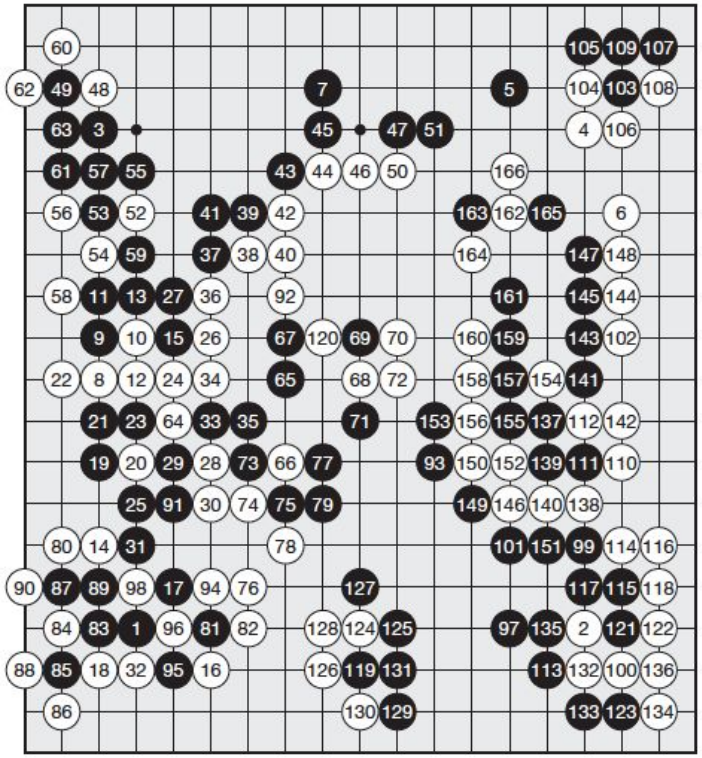
- 状態: ゲーム画面 (N x W x H)
- 行動: ボタン操作 (~10)
- 報酬: 各時刻のスコア増減

- 動きは決定的だが画面に出ない情報もある(det-POMDP)
- ルールが未知の状態から学習を開始

強化学習の具体例

- 目標：囲碁で勝つこと

- 状態: 盤面
- 行動: どこに石を置くか
- 報酬: 勝敗



(Silver, 2016)

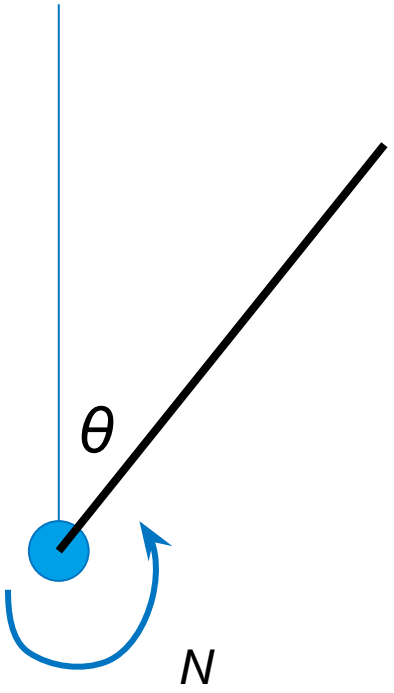
- ゲームのルールは学習前から分かっている
- 対戦相手の中は見えない
- 報酬は最終ステップのみ得られる

強化学習の具体例

- 目標：棒を逆さに立てる

- 状態: θ, ω
- 行動: トルクN
- 報酬: $\cos(\theta)$

- 状態と行動は連続的である



講義の流れ

1. 深層強化学習イントロダクション
2. 強化学習の問題設定
3. **価値関数ベースの手法**
4. 方策関数ベースの手法

価値関数

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots$$

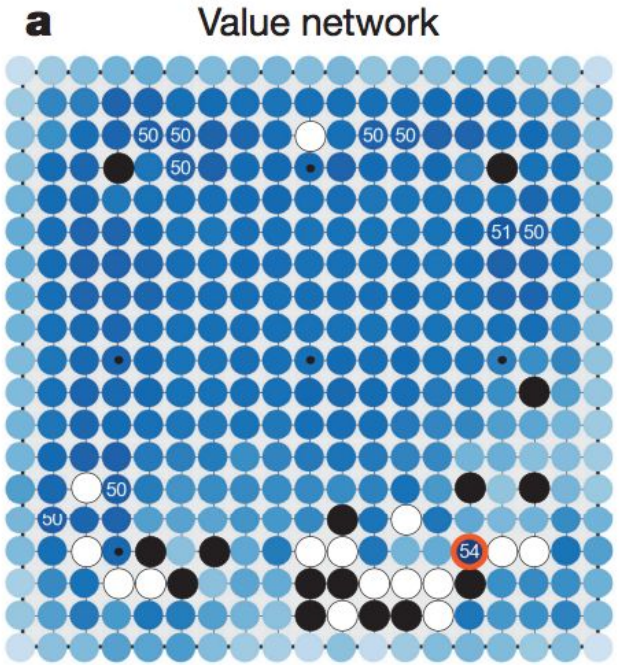
- 価値関数(Value function)
  - $v_\pi(s)$: 方策πのもとで、状態sから先の将来得られる、報酬の合計の期待値（利得Gの期待値）
  - $q_\pi(s, a)$: 同様に、状態sで行動aをとった先の利得期待値

  - 最適な方策の元での価値関数(v_*, q_*などと表記する)が分かれば、価値関数の値を大きくするような行動を選択していけば、それは最適な行動となる

価値関数の具体例

- 例1: AlphaGoの価値関数

  (元論文より)



(Silver, 2016)

- 例2: 将棋の価値関数(Bonanza)
  - V(S) = Σ[v(駒の価値)] + Σ[w(駒3つの位置関係)]

Q-learning

- 方策πの元でのQ関数

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

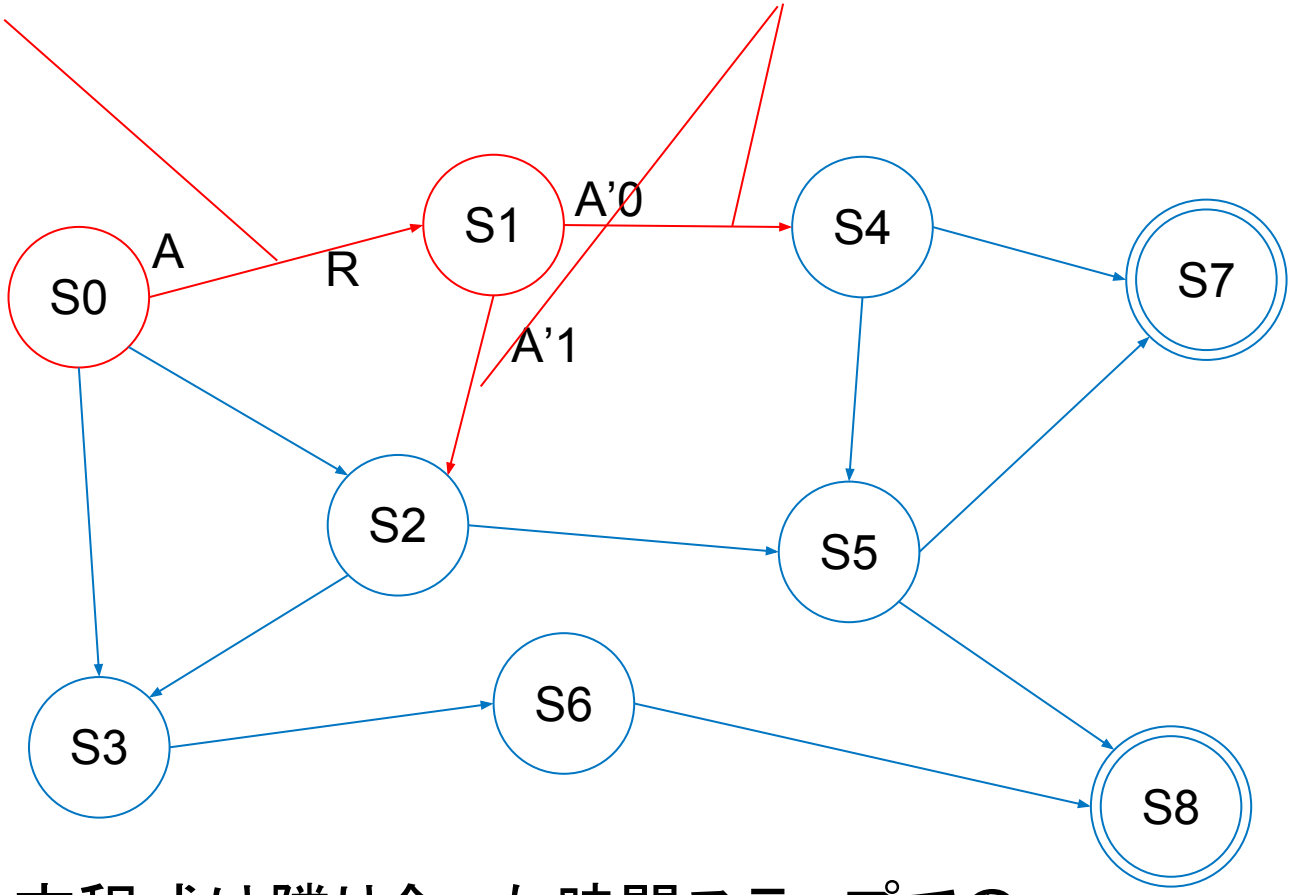
$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \;\middle|\; S_t = s, A_t = a\right]$$

- Bellman最適方程式
  - 最適な方策の元でのQ関数を考えると、この関数は次の Bellman最適方程式を満たす

$$q_*(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \;\middle|\; S_t = s, A_t = a\right]$$

即時の報酬　　γ×(時刻t+1以降の最適利得期待値)

$$q^*(S0, A) = R + \gamma \max_{A'}[q^*(S1, A')]$$



Bellman方程式は隣り合った時間ステップでの
Q関数の関係を表す式

Q-learning

$$q_*(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \,\middle|\, S_t = s, A_t = a\right]$$

TD学習

- Bellman最適方程式を満たすq*を求めたい
  - Bellman最適方程式の左辺を右辺に近づけるように反復
  - (S_t, A_t, R_{t+1}, S_{t+1})の経験から、次の更新を行う

$$\delta_t = (R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t))$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\delta_t$$

Q-learning

Exploration-Exploitation tradeoff

- ε-greedy探索
  - $(S_t, A_t, R_{t+1}, S_{t+1})$の経験を集めるための方策
    - 完全にランダムサーチだと非効率
    - 学習途中のQ関数に基づくgreedy policyは最適でない

  - 確率εでランダムなアクションを選ぶ
  - 確率1-εで現状のQ関数のもとで最適なアクションを選ぶ

$$\pi(S_t) = \underset{a'}{\operatorname{argmax}}[Q(S_t, a')] \quad \text{(確率1-ε)}$$

Q-learning

$$\delta_t = (R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t))$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \delta_t$$

- アルゴリズムまとめ

```
1  Initialize Q(s, a)
2  for each episode:
3      S ← S0
4      for each timestep:
5          Choose A from S using ε-greedy
6          Take action A, observe R, S'
7          update Q(S, A) based on (S, A, R, S')
8          S ← S'
```

DQN - 関数近似

- Q関数をどう表現するか？
  - →ナイーブには(s, a)の全ての組み合わせを配列に持つ
  - 囲碁：10^170状態
  - 100x100の2値画像：2^10000状態

とても対処できない

- 関数近似法

$$q(s,a) \approx \hat{q}(s,a;\theta)$$

  - ここでDeep Neural Networkで近似するのが

    Deep Q Network (DQN)

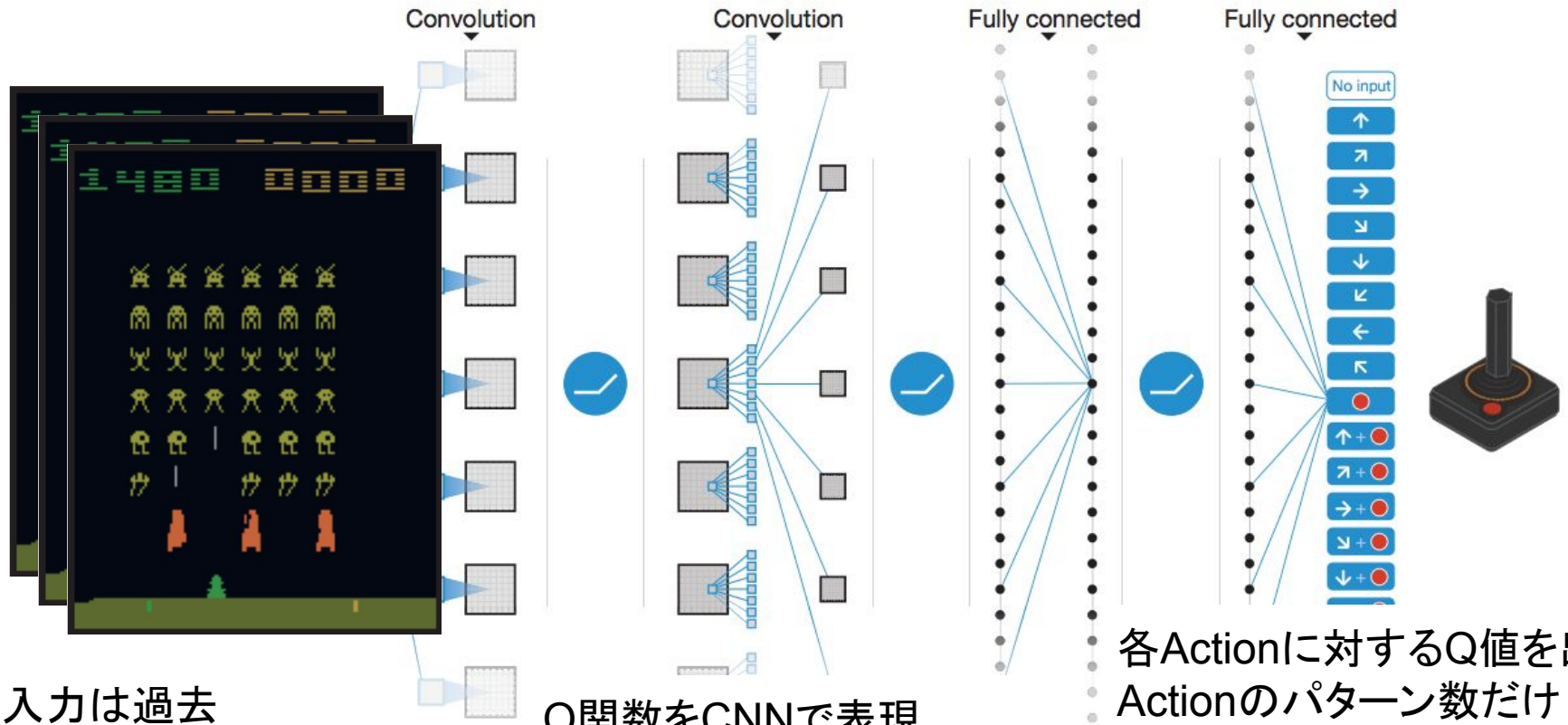
DQN - θの学習方法

- Semi-Gradient Q-learning
  - 「Bellman最適方程式の左辺を右辺に近づける」
  - 右辺 $y_t = R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a'; \theta)$
  - Loss関数 $L(\theta) = \mathbb{E}[(y_t - Q(S_t, A_t; \theta))^2]$
  - SGDなどで*L*を最小化する

  - yは本当はθに依存しているが、勾配を計算するときは ∂y/∂θはとらない (Semi-gradient)

DQN - playing Atari

- Atariのゲームプレイングを学習するモデル



入力は過去
数フレームの画像

Q関数をCNNで表現

各Actionに対するQ値を出力。
Actionのパターン数だけ
出力ユニットがある

(Minh, 2015)

DQN - playing Atari

DQN

- DQNを現実の問題に適用する例

DQN - playing Atari

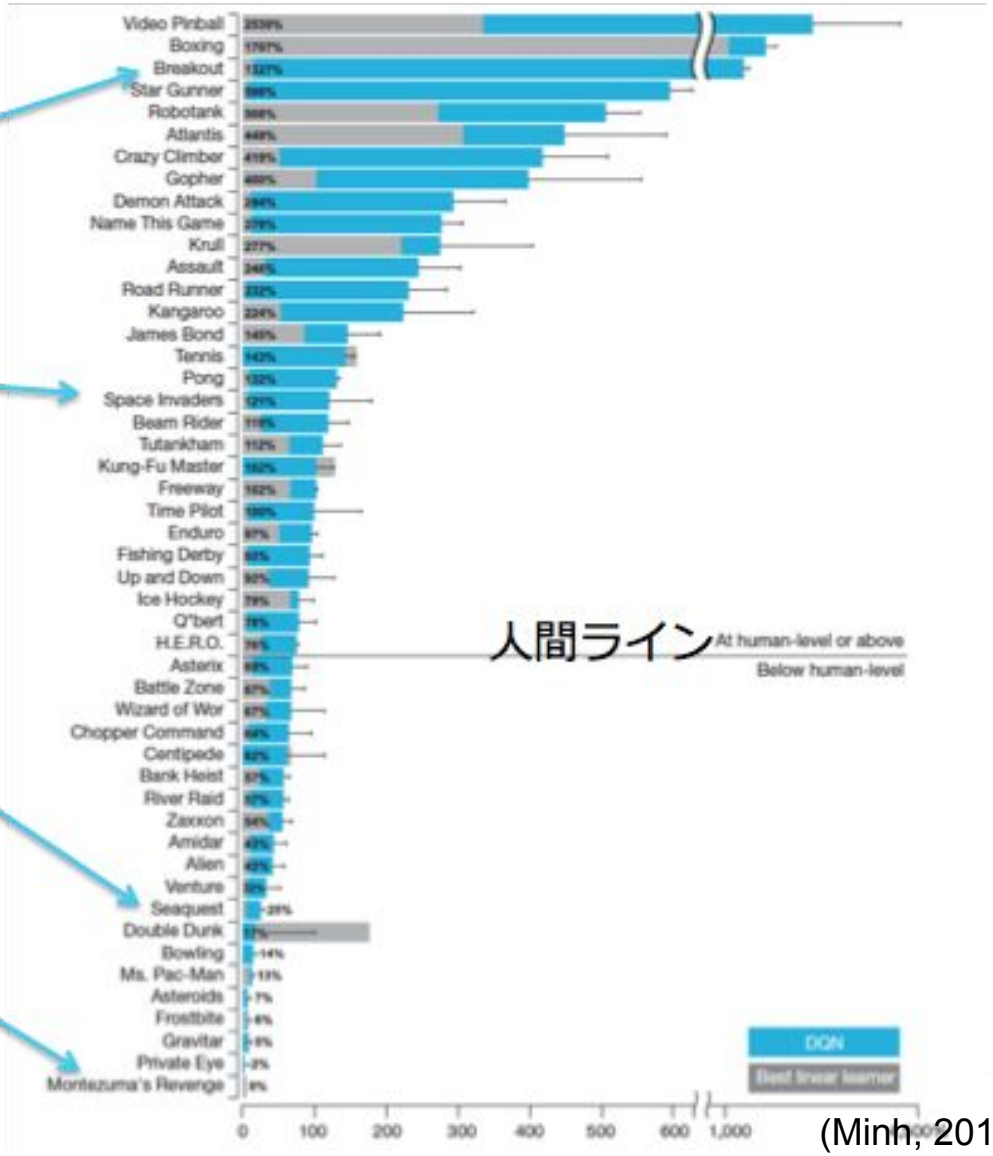
- DQNの性能評価

ブロック崩し

スペースインベーダー

Seaquest

Montezuma's Revenge
(ARPG)



(Minh, 2015)

Montezuma's Revenge

- DQNにとって難しい問題
  - 鍵を取って初めて0でない報酬が入る

Bellemare et al
2016

講義の流れ

1. 深層強化学習イントロダクション
2. 強化学習の問題設定
3. 価値関数ベースの手法
4. **方策関数ベースの手法**

方策関数ベースの手法

- 方策関数（確率的）

$$\pi(a|s;\theta)$$

- 方策勾配: 方策関数のパラメタθの変化が利得に及ぼす影響

$$J(\theta) = \mathbb{E}[G_0]$$

$$\nabla_\theta J(\theta) = \mathbb{E}_s[\sum_a Q^\pi(s,a)\nabla_\theta \pi(a|s;\theta)]$$

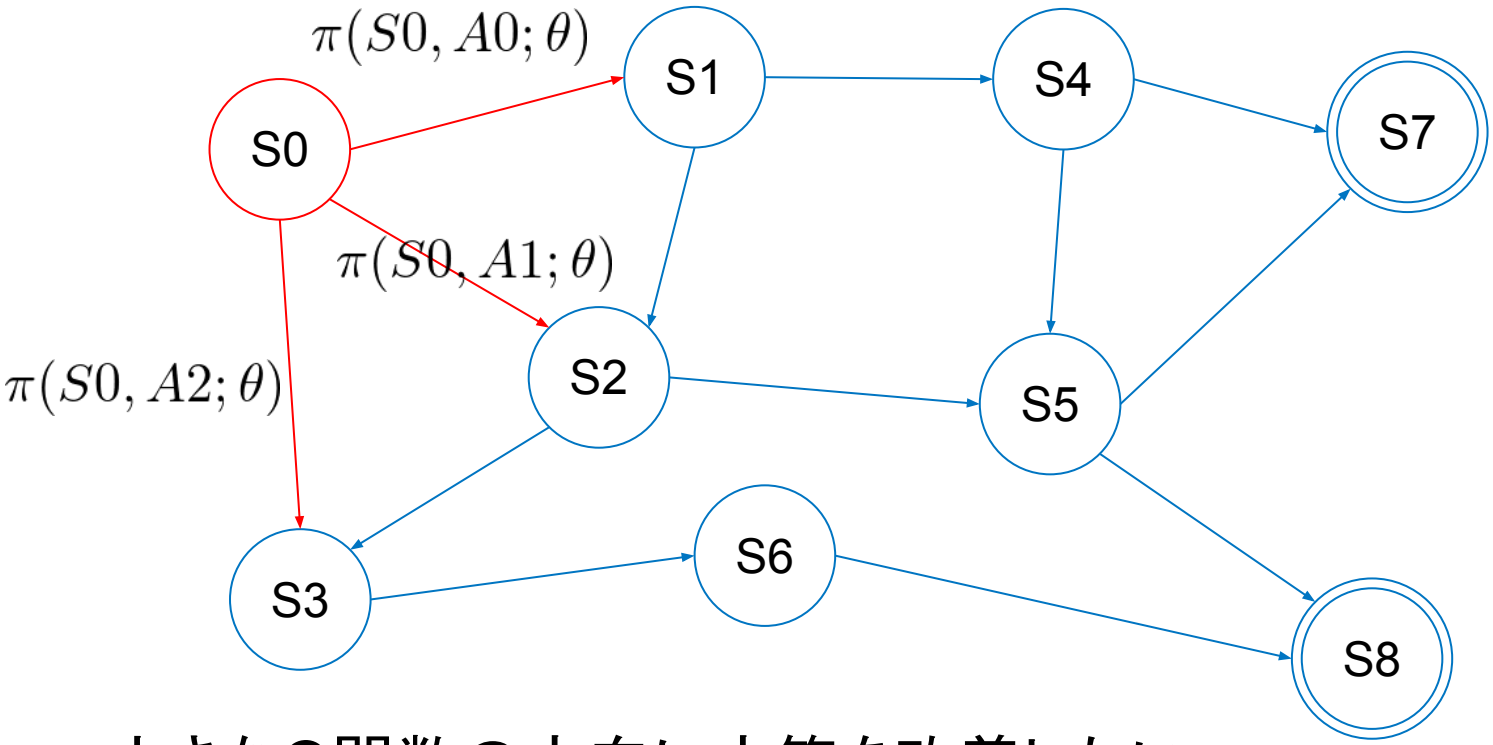
$$= \mathbb{E}_{(s,a)}[Q^\pi(s,a)\nabla_\theta \log(\pi(a|s;\theta))]$$

- この勾配にしたがってθを最適化するアルゴリズム

※ちゃんとした導出は12月の授業で行う

$$\nabla_\theta J(\theta) = \mathbb{E}_s[\sum_a Q^\pi(s,a)\nabla_\theta \pi(a|s;\theta)]$$

NNでモデル化してればBack propagation可能



大きなQ関数の方向に方策を改善したい

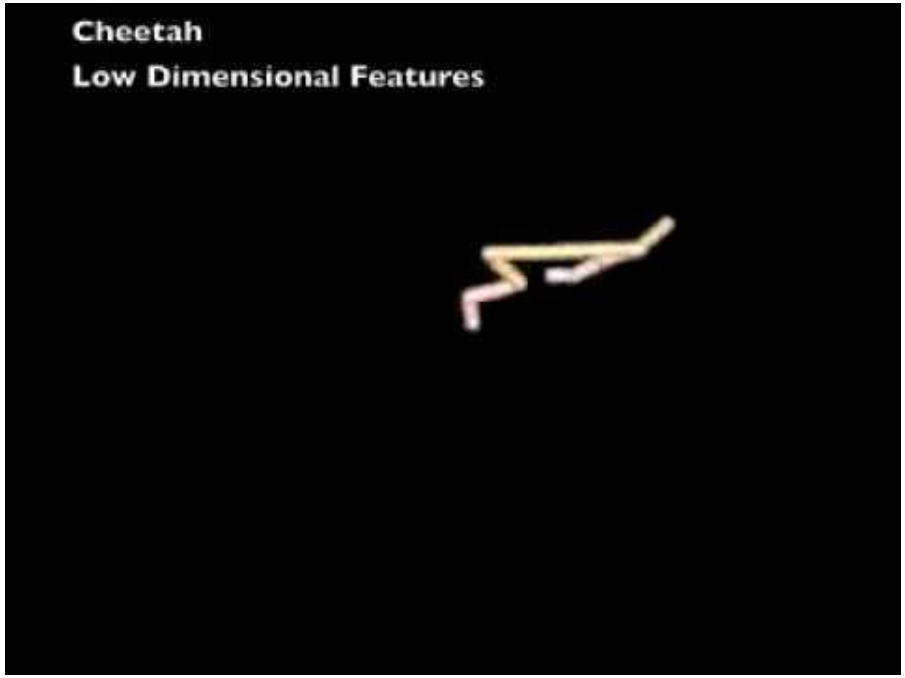
方策勾配の計算方法

- Qをモンテカルロ法で近似 …… REINFORCE
- QをTD学習 …… Actor-Critic法
- 派生アルゴリズムがたくさん
  - 自然勾配法にしたもの … Natural Actor-Critic
  - Qの代わりにA=Q-Vを使ったもの … Advantage Actor-Critic
  - 決定的なPolicyを使うもの … Deterministic Policy Gradient
  - などなど

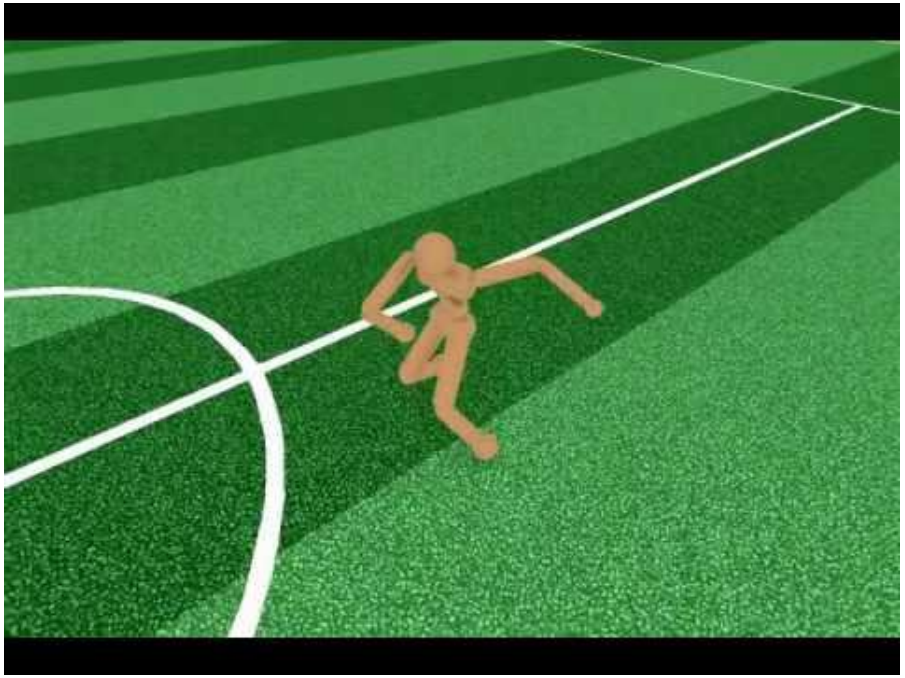
方策関数ベース手法の有効性

- 次のようなときに方策関数ベースの手法が有効である
  - 行動が高次元 or 連続的なとき(DQNは離散action)
  - 方策関数の形がある程度分かっているとき
    - 関数形が既知（周期関数など）
    - 教師データがある（プロ棋士の棋譜など）
    - 別のタスクで学習したものから転移できる

方策関数ベース手法の具体例



Lillicrap+ 2015



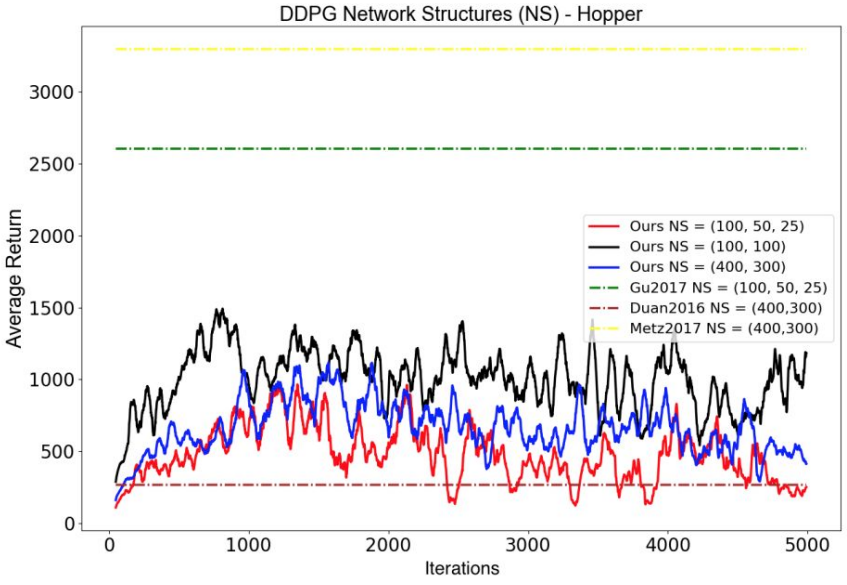
Schulman+ 2017

まとめ

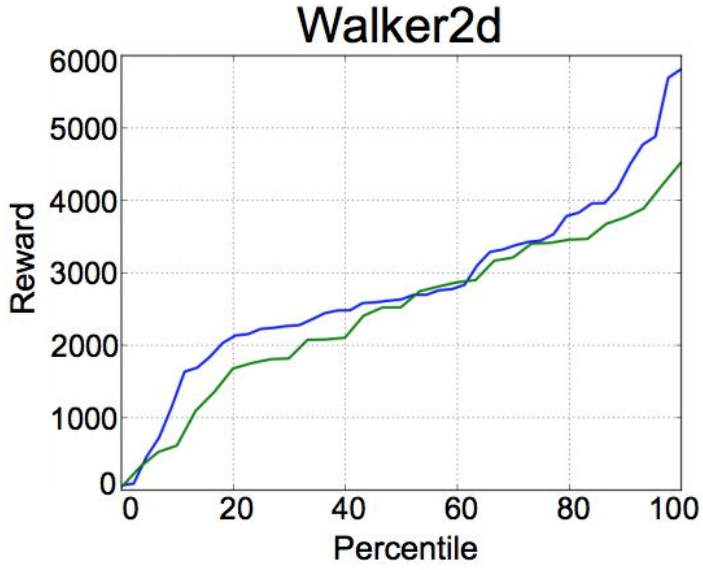
1. 深層強化学習イントロダクション
2. 強化学習の問題設定
3. 価値関数ベースの手法 → 11/21の講義で詳しく扱う
4. 方策関数ベースの手法 → 12/19の講義で詳しく扱う

深層強化学習の抱える課題

- 学習に時間がかかる
  - >1M timestep
- 学習が不安定でハイパーパラメタ調製が難しい

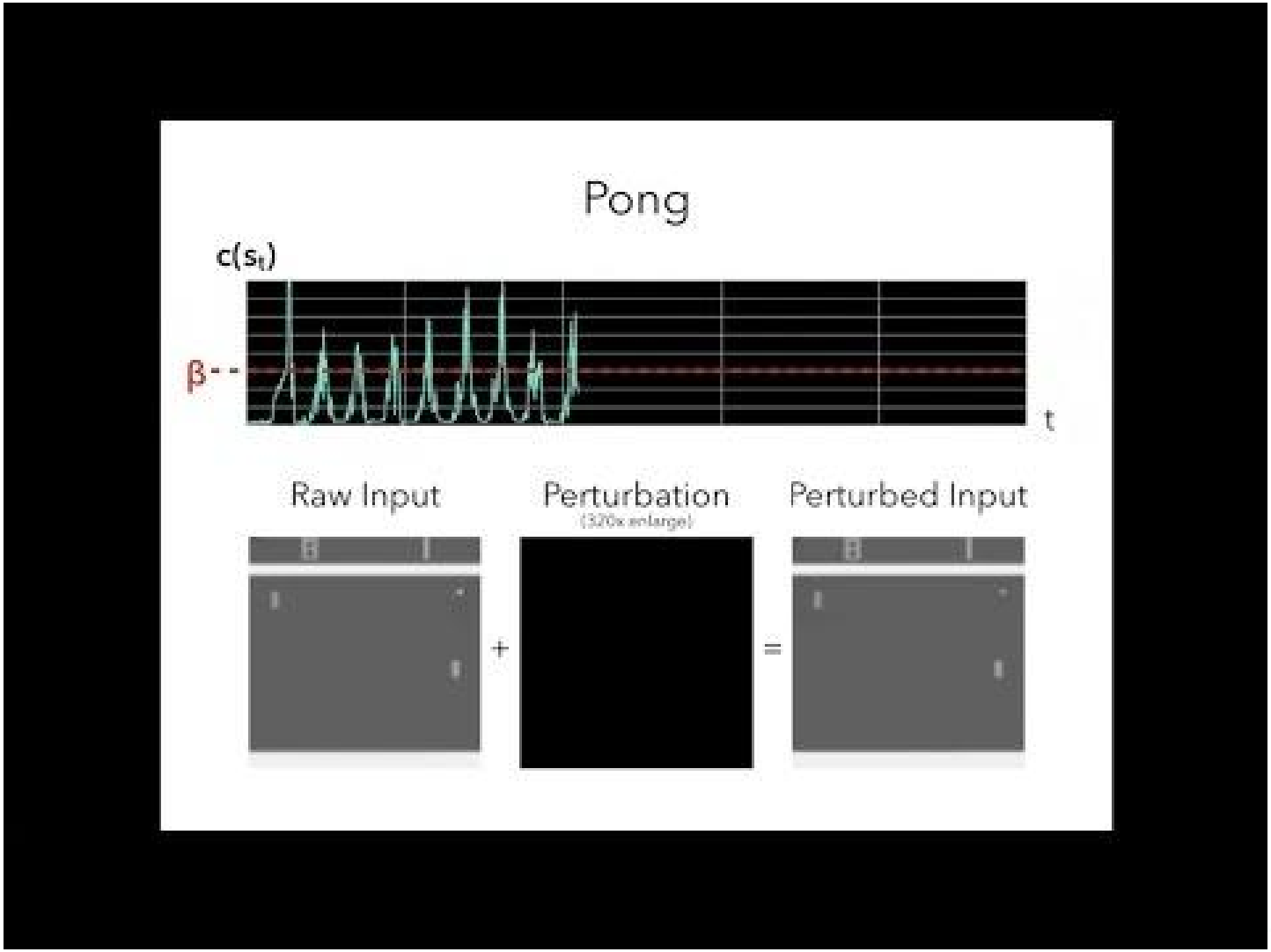


Islam+, 2017

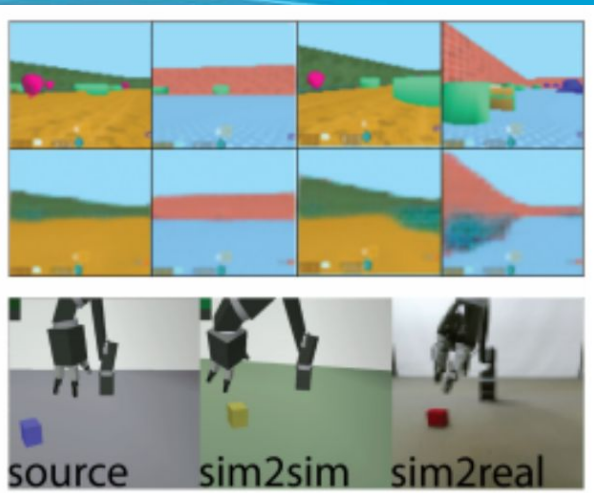


Pinto+, 2017

深層強化学習の抱える課題

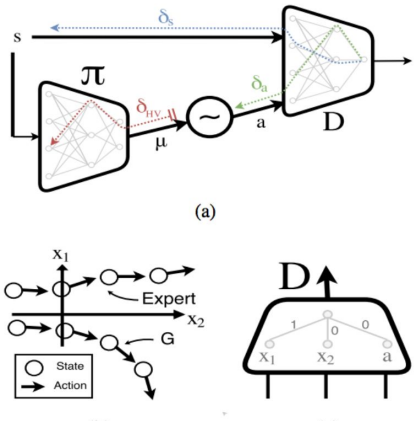


ホットな研究テーマ

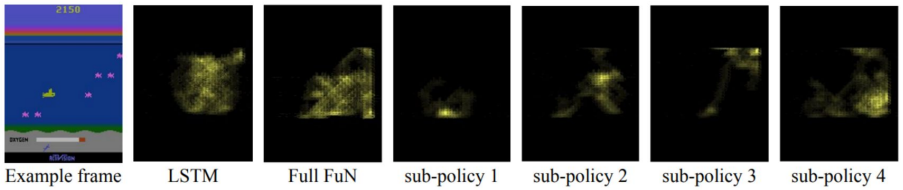


https://arxiv.org/pdf/1707.08475.pdf

- 学習の安定化
- 学習の高速化
- 転移学習（特にシミュレーションから実機）
- 予測モデルの活用
- 探索の効率化
- 模倣学習
- 階層強化学習



http://proceedings.mlr.press/v70/baram17a/baram17a.pdf



https://arxiv.org/pdf/1703.01161.pdf

参考文献

- David Silverの講義
  - http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
- Richard S. SuttonとAndrew G. Bartoの教科書
  - https://webdocs.cs.ualberta.ca/~sutton/book/bookdraft2016sep.pdf

参考文献

- Atari
  - Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602 (2013).
  - Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533.
- AlphaGo
  - Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489.

参考文献

- Y. Li, □Deep reinforcement learning: An overview,□ arXiv preprint arXiv:1701.07274, 2017.
- L. Pinto, et al., □Robust adversarial reinforcement learning,□ 2017.
- L. Zoph, □Neural architecture search with reinforcement learning,□Proc. ICLR, 2017.
- A. S. Vezhnevets, et al., □Feudal networks for hierarchical reinforcement learning,□ in Proc. ICML, 2017.
- Bellemare, Marc G., et al. "Unifying Count-Based Exploration and Intrinsic Motivation." *arXiv preprint arXiv:1606.01868* (2016).
- Lillicrap, Timothy P. et al. “Continuous Control with Deep Reinforcement Learning.” In *ICLR*, 2016.
- Gu, Shixiang et al. “Continuous Deep Q-Learning with Model-Based Acceleration.” In *ICML*, 2016
- Mnih, Volodymyr et al. “Asynchronous Methods for Deep Reinforcement Learning.” In *ICML*, 2016.
- Hausknecht, Matthew, and Peter Stone. “Deep Recurrent Q-Learning for Partially Observable MDPs.” In *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (AAAI-SDMIA15)*, 2015
- Oh, Junhyuk et al. “Control of Memory, Active Perception, and Action in Minecraft.” In *ICML*, 2016
- Hasselt, Hado van. “Double Q-Learning.” In *NIPS*, 2010.
- Hasselt, Hado van, Arthur Guez, and David Silver. “Deep Reinforcement Learning with Double Q-Learning.” In *AAAI*, 2016.
- Bellemare, Marc G. et al. “Increasing the Action Gap: New Operators for Reinforcement Learning.” In *AAAI*, 2016.
- Nair, Arun et al. “Massively Parallel Methods for Deep Reinforcement Learning.” In *ICML Deep Learning Workshop*, 2015.
- Stadie, Bradly C., Sergey Levine, and Pieter Abbeel. “Incentivizing Exploration In Reinforcement Learning With Deep Predictive Models.” arXiv preprint arXiv:1507.00814 (2015).
- Oh, Junhyuk et al. “Action-Conditional Video Prediction Using Deep Networks in Atari Games.” In *NIPS,* 2015.
- Bellemare, Marc G., et al. "Unifying Count-Based Exploration and Intrinsic Motivation." *arXiv preprint arXiv:1606.01868* (2016).
- Kulkarni, Tejas D., et al. "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation." *arXiv preprint arXiv:1604.06057* (2016).

参考文献

- Game AI video
  - https://www.youtube.com/watch?v=iqXKQf2BOSE
  - https://www.youtube.com/watch?v=5WXVJ1A0k6Q
  - https://www.youtube.com/watch?v=0yI2wJ6F8r0
  - https://youtu.be/0yI2wJ6F8r0
  - https://youtu.be/jQg8p-V8jF4
- Robot video
  - https://www.youtube.com/watch?v=ATvp0Hp7RUI
  - https://youtu.be/tJBIqkC1wWM
  - https://www.youtube.com/watch?v=1bKYLoskSCM&nodirect=1
- PFN video
  - https://www.youtube.com/watch?v=a3AWpeOjkzw
  - https://www.youtube.com/watch?v=7A9UwxvgcV0
  - https://youtu.be/yFCCanSxOE4
  - https://youtu.be/MpWvJhznpQQ (これは強化学習ではない)